PVFS2

PARALLEL VIRTUAL FILE SYSTEM

Birds of a Feather

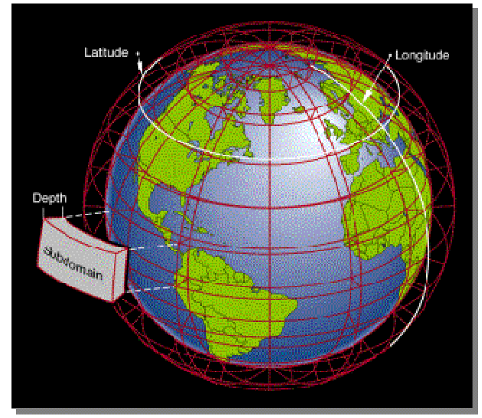
SC2005

Seattle, WA

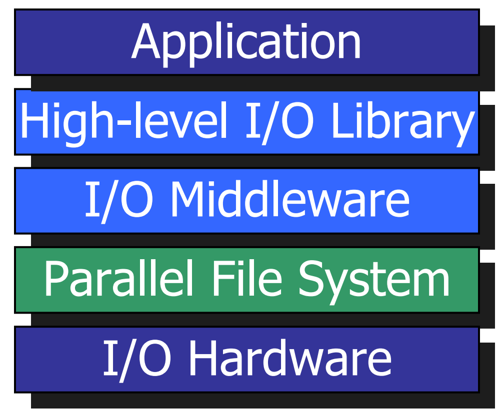
http://www.pvfs.org/pvfs2

HEC I/O Software Stacks

- Computational science applications have complex I/O needs
  - Performance and scalability requirements
  - Usability (Interfaces!)
- Software layers combine to provide functionality
  - High-level I/O libraries provide useful interfaces
    - Examples: Parallel netCDF, HDF5
  - I/O Middleware optimizes and matches to file system
    - Example: MPI-IO
  - Parallel file system organizes hardware and actually moves data



Graphic from J. Tannahill, LLNL

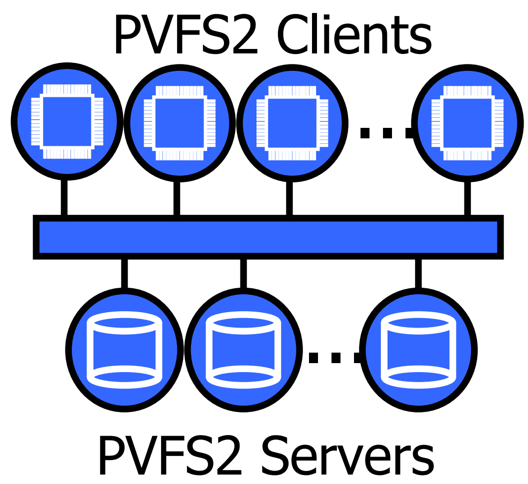


Role of Parallel File Systems

- **Manage storage hardware**
  - Present a single logical view of lots of components
  - Provide data redundancy for some environments
- **Scale to very large numbers of clients**
  - Handle many concurrent, independent accesses
  - Extract as much of the hardware performance as possible
  - Consider client failures to be a common case
- **Provide building-block API and semantics**
  - PFS is only one part of the I/O system
  - Must hook efficiently to MPI-IO implementation
- **Not…**
  - Replace your home directory file system
  - Store billions of tiny files
  - Be "global"

PVFS2 at a Glance

- **“Intelligent” servers**
  - Single server type
  - Can manage metadata, data, or both
  - Present a global file name space
- **Messaging over existing communication network**
  - Leverage that expensive network (e.g. GigE, Myrinet, IB)
  - Protocol tuned for HEC applications



- **Storage on disks locally attached to servers**
  - Both data and metadata may be distributed across servers
  - File data striped across servers for performance
  - Possibly shared access for failover purposes
- **MPI-IO and VFS interfaces for clients**
  - VFS for utilities, MPI-IO for applications
- **Open source, open development, and available for free**
  - Single CVS source tree with anonymous access
  - Mailing lists for user and developer discussion
  - Good vehicle for research in parallel I/O, in addition to production use

http://www.pvfs.org/pvfs2

Collaborators!

- Core development
  - Argonne National Laboratory
    - Ross, Lang, Latham, Vilayannur, Gropp, Thakur, Beckman, Coglan, Yoshii, Iskra
  - Clemson University
    - Ligon, Settlemyer
  - Ohio Supercomputer Center
    - Wyckoff, Baer
- Collaborators
  - Northwestern University
    - Choudhary, Ching
  - Ohio State University
    - Panda, Yu
  - University of Heidelberg
    - Ludwig, Kunkel
  - Acxiom Corporation
    - Carns, Metheny
  - Penn State University
    - Sivasubramaniam, Kandemir
  - University of Michigan
    - Honeyman, Hildebrand

University of Heidelberg

Acxiom Corporation

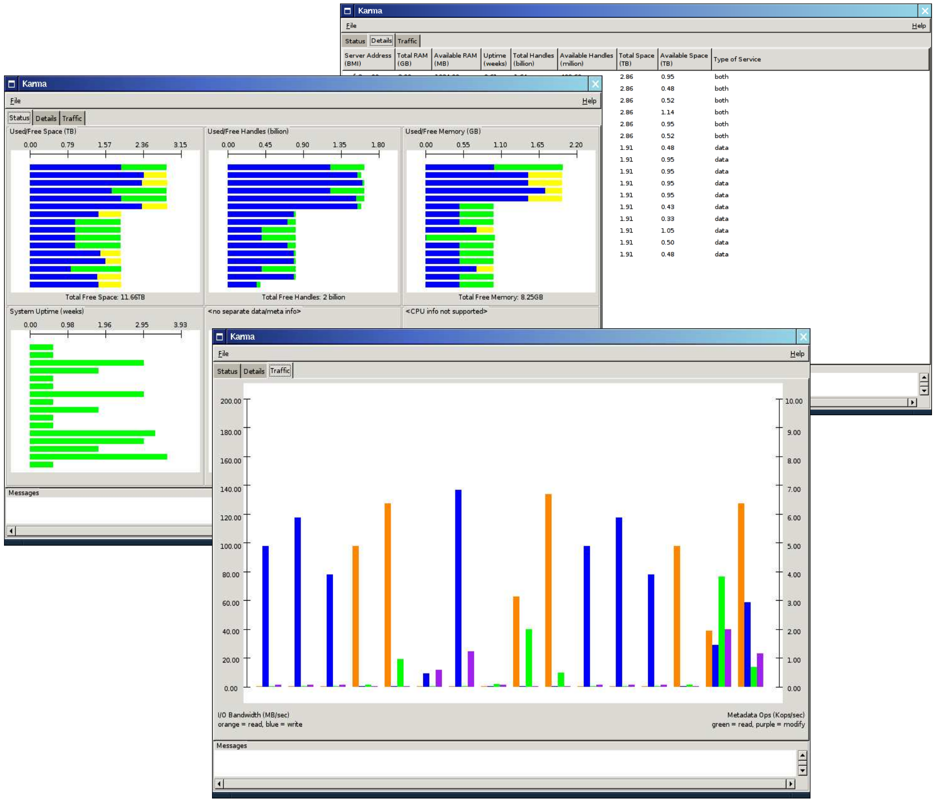
- "… the world's largest processor of consumer data …"
  – Fortune magazine
- Multi-national (US, UK, France, Australia, Japan)
- Houses data and runs analytics applications for financial and other large businesses
  - Compute and data intensive
  - 24/7 operation
  - Highly-available, redundant resources
  - 7000+ compute nodes deployed in widely distributed environment

Acxiom and PVFS

- Deploy PVFS1 as data storage solution
  - 80 PVFS clusters
    - 16 nodes/cluster
    - Data loosely mirrored between clusters for redundancy
  - 750TB+ of PVFS storage deployed so far
  - Many internal applications use PVFS libraries directly
- Actively participating in PVFS2 development
  - Hired Phil Carns ☺
  - Evaluating PVFS2 as replacement for existing PVFS deployment
  - Interest in highly available PVFS2 clusters

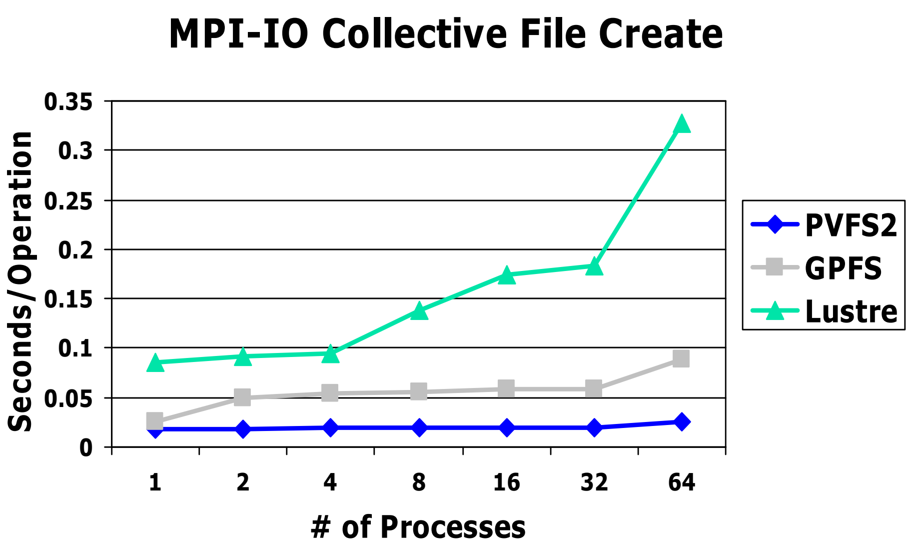
Accessibility and Management

- Doesn’t take a rocket scientist to install
  - One server type, simple configuration
- Can be re-exported via NFS for access on other OSes
- GUI and command line tools for monitoring and administration
  - pvfs2-fsck for salvaging damaged file systems

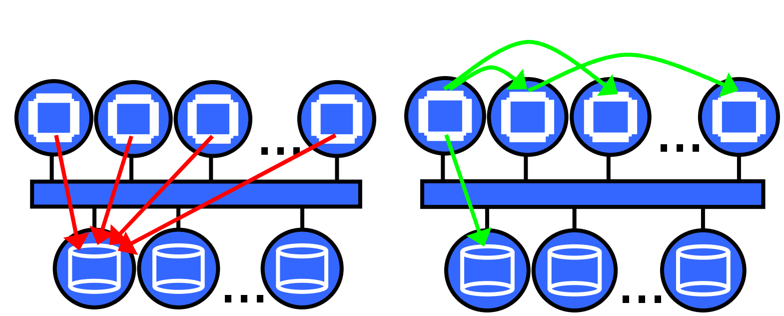


Scalable MPI-IO

- MPI-IO is a critical component of the HEC I/O stack
  - Used both directly and indirectly by HEC applications
- Without tuning FS can see storm of calls for MPI-IO operations such as open, create, and resize
- PVFS2 and ROMIO MPI-IO implementation work together to make these operations efficient and scalable
- Noncontiguous I/O advances in PVFS2 and ROMIO as well…

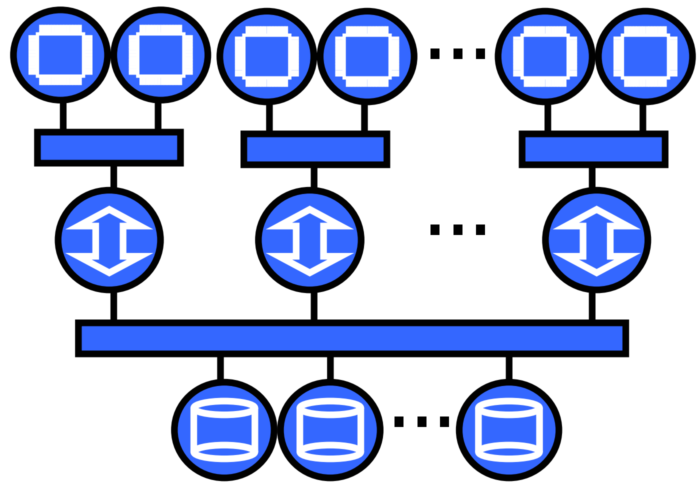


PVFS2 results from OSC; Lustre and GPFS from LLNL



Unoptimized MPI-IO create and open operations require communication from every process to the file system. With optimizations, communication between processes and file system is minimized.

Very Large Scale Systems



- PVFS2 and ROMIO MPI-IO are designed to meet the demands of this scale
  - I/O aggregation, statelessness, lockless solutions, rich semantics, and failure tolerance
- (Some of) these systems have support nodes between processors and storage
  - Building software (with others) to efficiently handle I/O forwarding from processors to storage
  - Hoping this approach will be adopted across the board

PVFS2 is ready for use.

- Runs on a wide variety of architectures, Linux kernels, and interconnects
- Running in production, actively tested by many users
  - Averaging 130+ downloads per month
  - 156 PVFS2-users subscribers
- Performance will improve, but it isn't bad now (multi-GB/sec large I/O)
- Documentation, support mailing lists, etc. are all in place
- Failover, pvfs2-fsck, and GUI monitoring help complete the package

Next…

- Troy Baer (Ohio Supercomputer Center)
  PVFS2 deployment at OSC
- Avery Ching (Northwestern Univ.)
  Noncontiguous I/O, ROMIO, and PVFS2
- Dean Hildebrand (Univ. of Michigan)
  NFSv4, pNFS, and PVFS2
- Rob Latham (Argonne)
  PVFS2 on IBM BlueGene/L systems

- Open discussion

PVFS2 in Production at OSC

Troy Baer
Science & Technology Support Group
Ohio Supercomputer Center

PVFS2 in Production at OSC

- OSC's Columbus HPC environment
  - HPC systems
  - Mass storage environment
- PVFS2 at OSC
  - Configuration
  - Performance
  - Trials and Tribulations
  - Applications

OSC's Columbus HPC Environment

- Pentium 4 cluster
  - 112 dual 2.4GHz nodes with InfiniBand and Gigabit Ethernet
  - 144 dual 2.4GHz nodes with Gigabit Ethernet
- Itanium 2 cluster
  - 128 dual 0.9GHz nodes with Myrinet and Gigabit Ethernet
  - 110 dual 1.4GHz nodes with Myrinet and Gigabit Ethernet
  - 20 dual 0.9GHz nodes with Gigabit Ethernet
  - 1 32-way 1.3GHz SGI Altix 3700
  - 3 16-way 1.4GHz SGI Altix 350s

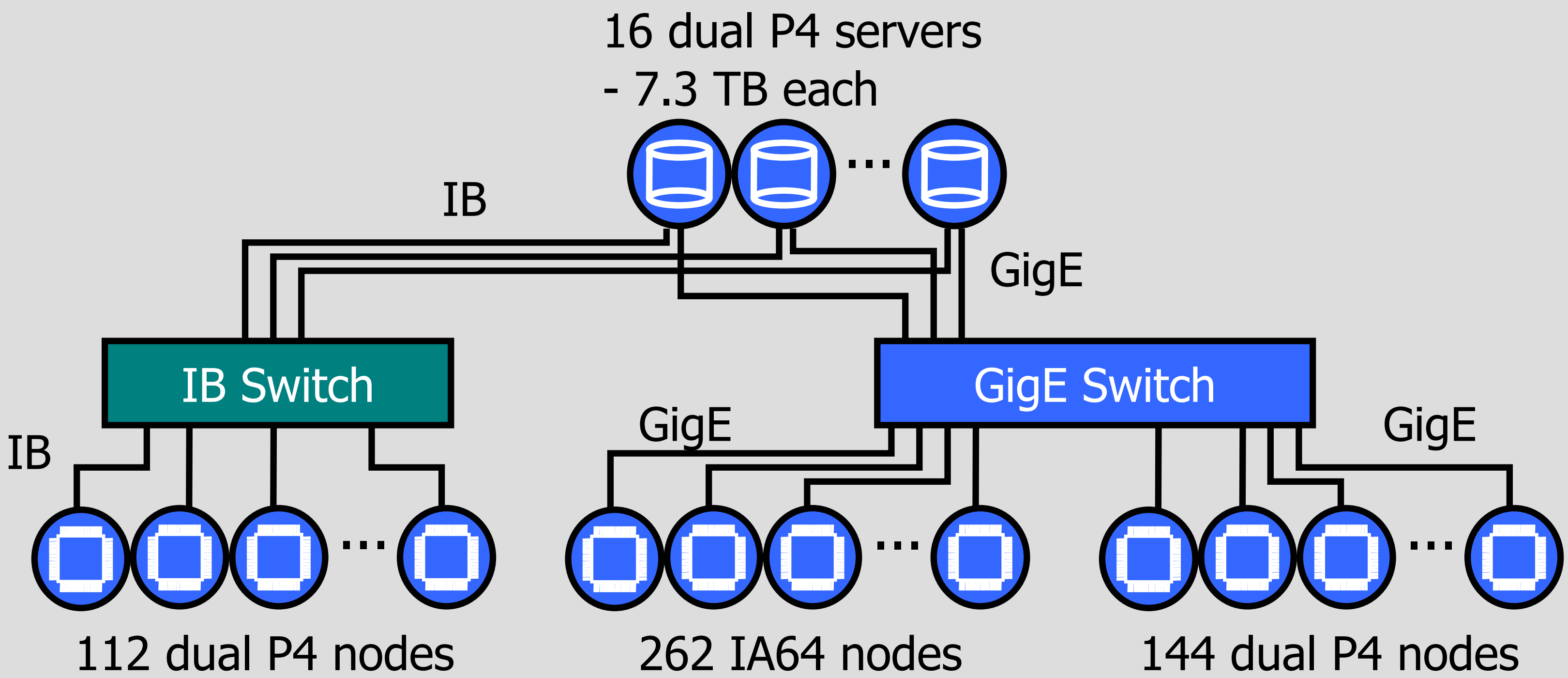
OSC's Columbus HPC Environment (con't.)

- Mass Storage Environment
  - 4 IBM FAStT900s with 12 TB of FC disks each
  - 20 IBM FAStT600s with 20 TB of SATA disks each
  - IBM 3584 tape library with 4 drives and ~80 TB of LTO tapes
  - 7 NFS servers for user home directories (~56 TB total)
  - 2 TSM backup servers
  - ***<u>16 PVFS2 servers (~116 TB total)</u>***

PVFS2 at OSC -- Goals

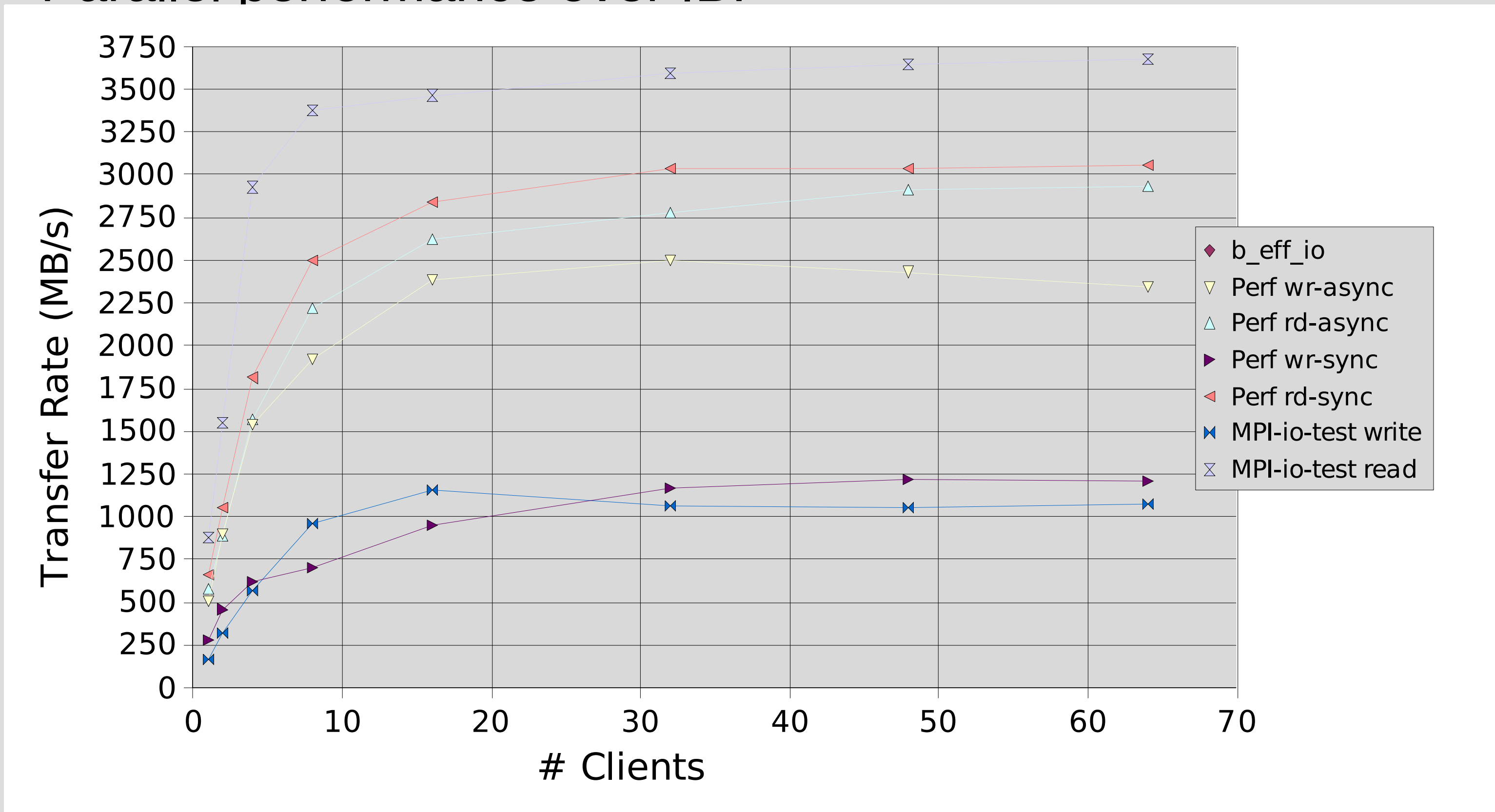
- High performance
  - Equal or better than local disk for serial applications for very large files
  - Much better than NFS for parallel applications
  - Use high performance network (eg. IB) where available
- High capacity
  - Much larger than any single node's local disk
  - Allow for very data-intensive applications
- Shared
  - Across multiple architectures where possible

PVFS2 at OSC -- Configuration



PVFS2 at OSC -- Performance

- Serial performance over GigE:
  - 96.1 MB/s read (4 MB buffer size)
  - 90.9 MB/s write (4 MB buffer size)
- Parallel performance over IB:



Trials and Tribulations

- Not all software related!
  - GigE network topology was suboptimal for parallel I/O, particularly from IA64 nodes
  - Fixed by an upgrade of core GigE switch in October
- Most software problems have been with kernel VFS driver
  - Some vendors still only support Linux 2.4
  - Ugliest problems are those where the kernel driver cannot be unloaded once loaded

PVFS2 at OSC -- Applications

- The killer app is computational chemistry... but not necessarily the way you might expect.
  - Serial Gaussian jobs generating 300+ GB scratch files – too big to fit on any node's local disk!
  - Several independent research groups
- Another heavy user is CS research
  - Indexing metadata from HDF5 files in parallel

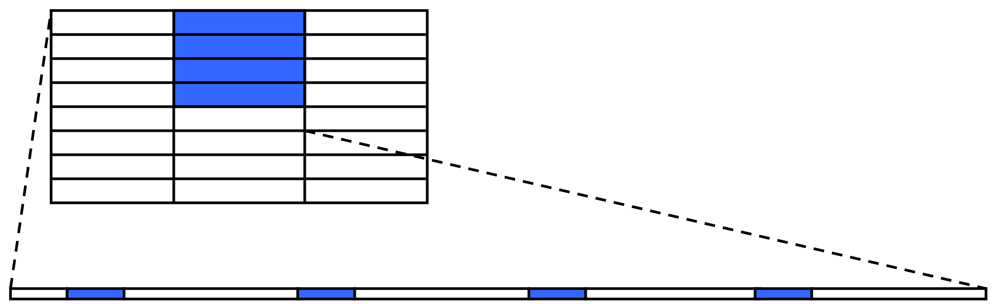
Research in Noncontiguous I/O

Avery Ching
Northwestern University

http://www.pvfs.org/pvfs2

Noncontiguous I/O

- File views and MPI datatypes allow users and libraries to describe noncontiguous I/O



- PVFS2 allows these regions to be described concisely as well
- ROMIO MPI-IO implementation can convert MPI-IO descriptions into PVFS2 ones
  - Current released version is not optimal

POSIX I/O

- Most file systems support this interface
- Break noncontiguous I/O operations into multiple contiguous I/O operations
- Often generates large number of I/O operations for noncontiguous I/O
- Small I/O operations is inefficient for hard disks

Two-phase I/O

- A collective I/O optimization built on the POSIX I/O interface
- I/O aggregators split file into domains and use data sieving I/O for all I/O operations
- Improves on data sieving by eliminating overlapping file regions and consistency control
- Overheads include passing data over the network twice, accessing some unused file data and in the write case r-m-w sequence cost

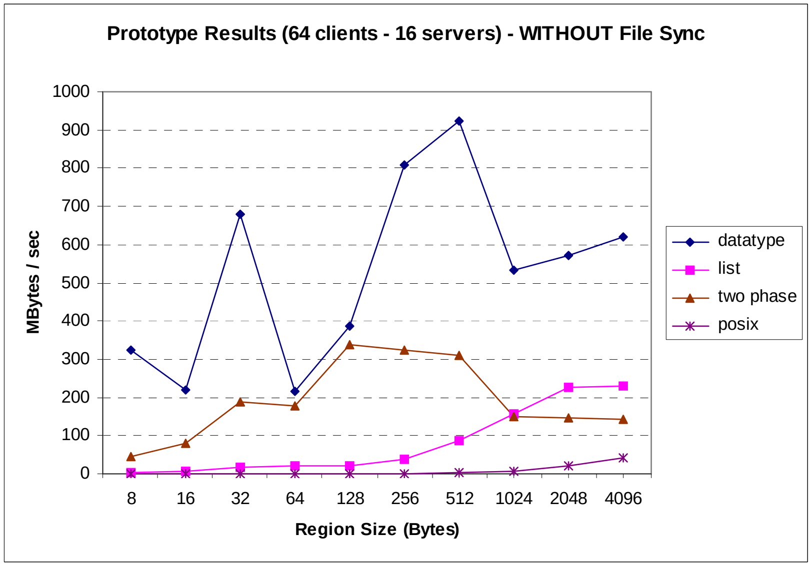
List I/O

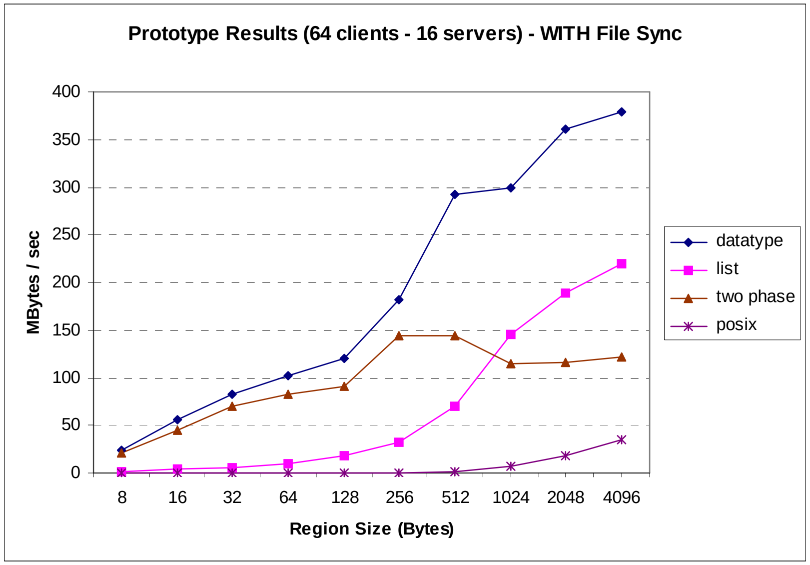
- A file system interface optimization
- A single I/O operation can handle noncontiguous I/O access patterns
- Overhead of passing file offsets and lengths across network on an I/O operation
- I/O operations split up after a fixed number of offset-length pairs

Datatype I/O

- A file system interface optimization
- Use derived datatypes to handle noncontiguous I/O
- No additional network bandwidth for increased region counts
- An efficient interface for structured data access

Prototype Results





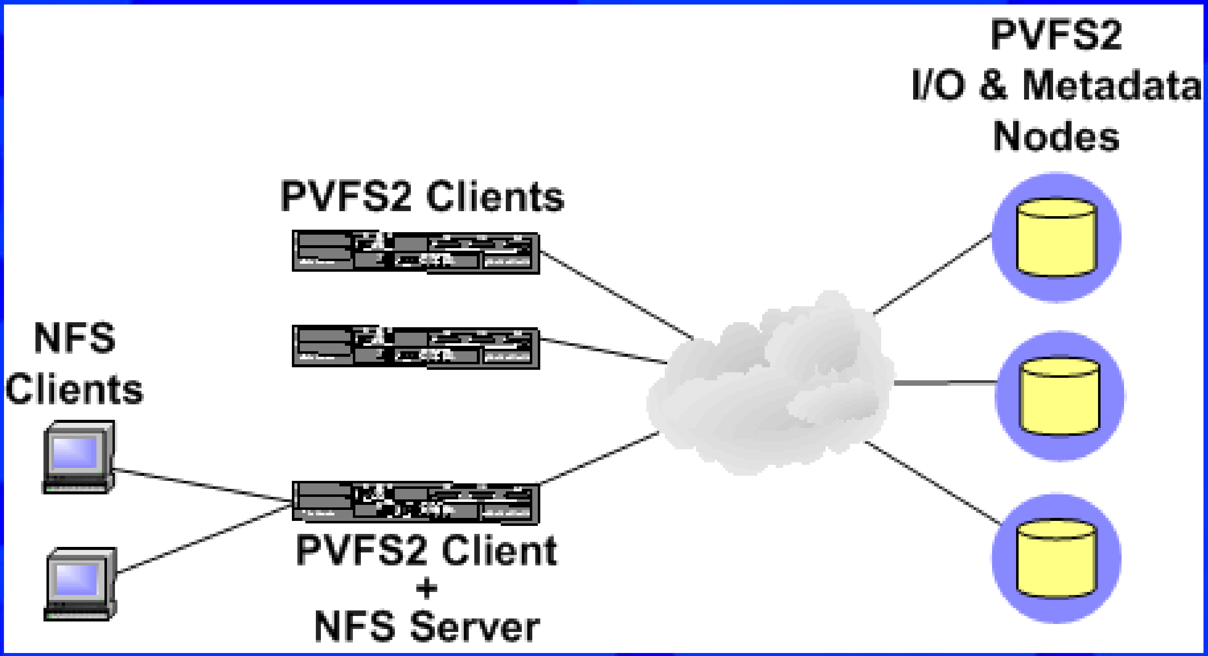
- Noncontiguous I/O implementations significantly affect performance
- Prototype MPI-IO driver for PVFS2 will be eventually integrated into ROMIO
- Automatic hint selection and performance tuning are necessary before integration

NFSv4, pNFS, and PVFS2

Dean Hildebrand

Advisor: Peter Honeyman
Center For Information Technology Integration
University of Michigan

NFS and PVFS2

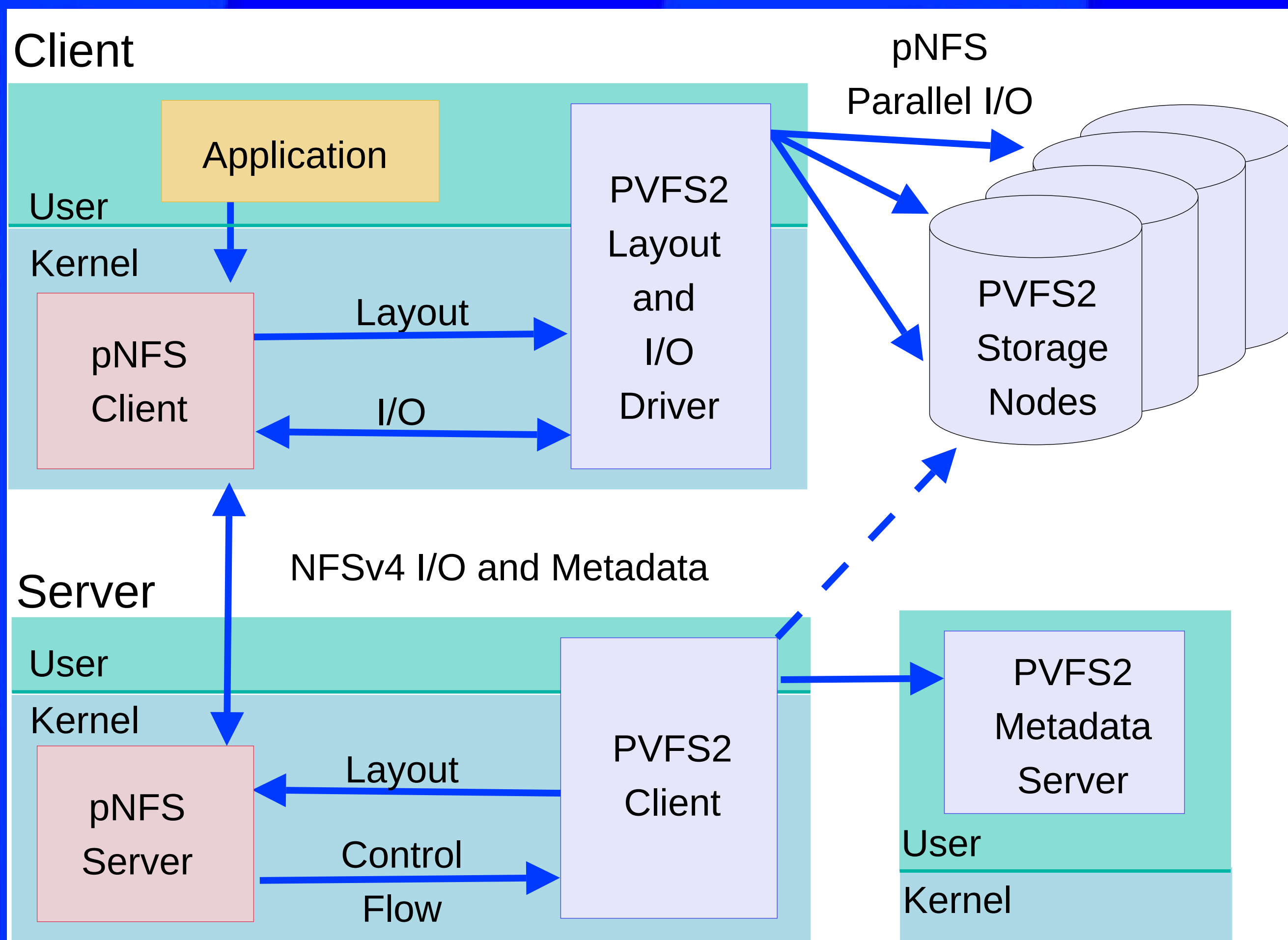


- NFS server exports PVFS2 kernel module
- 32KB requests and indirection reduces performance
- Increase NFSv3 scalability by increasing number of NFSv3 servers
- NFSv4 server maintain state information

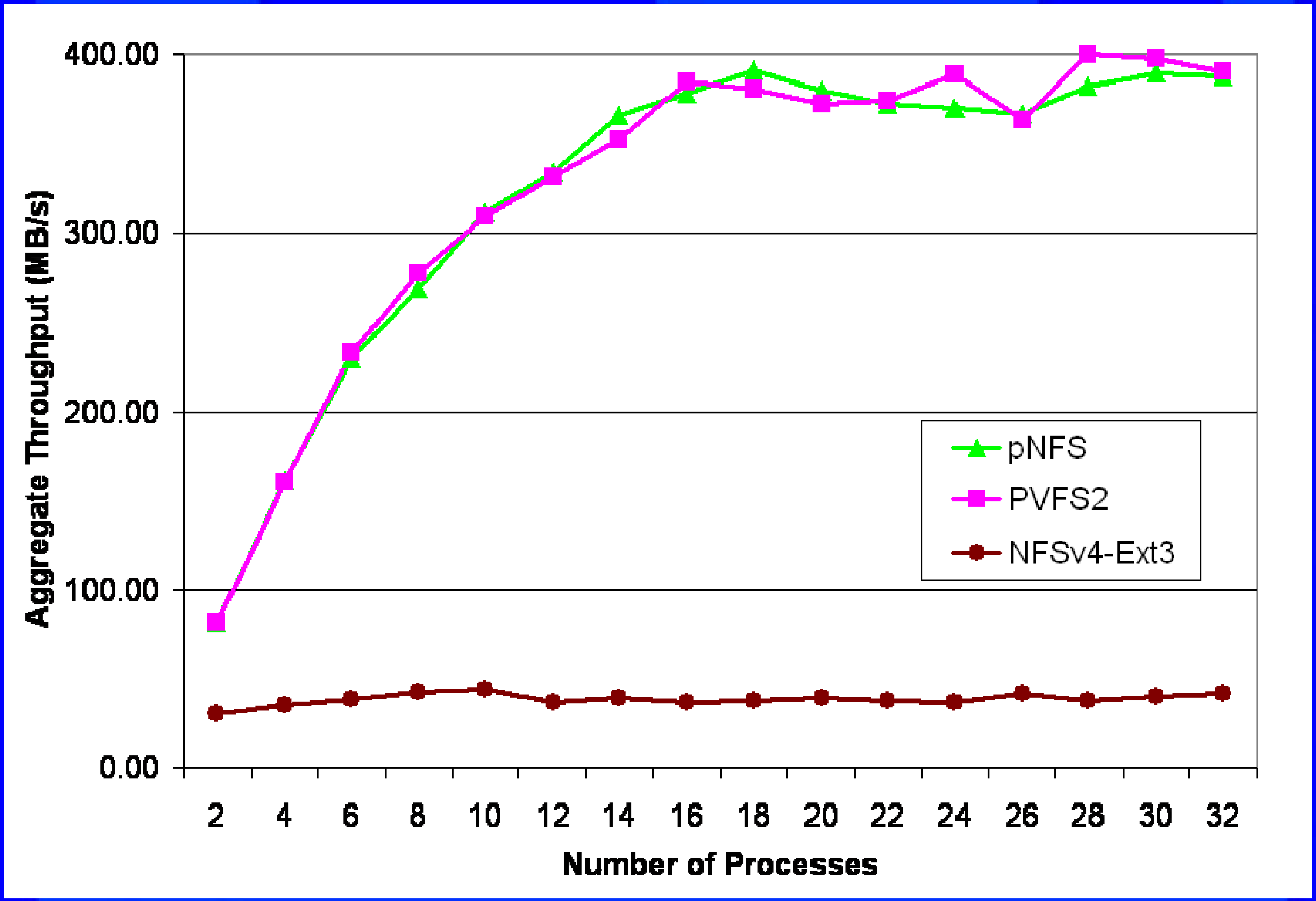
pNFS

- IETF NFSv4.1 protocol extension
- Scale with underlying file system
  - Clients performs direct I/O to storage
  - Escape NFSv4 block size restrictions
  - Single file access
- File system independent
  - Support all layout maps (block, object, file, etc)
  - Create global namespace of disparate HPC file systems
- Interoperate with standard NFSv4 clients and servers
  - Storage still accessible through NFSv4 server
- Operate over any NFSv4 infrastructure
- Support existing storage protocols and infrastructures
  - Examples: SBC on Fibre Channel on iSCSI, NFSv4
- Current "industry" support:
  - Sandia, LANL, Netapp, Sun, IBM, Panasas, NEC

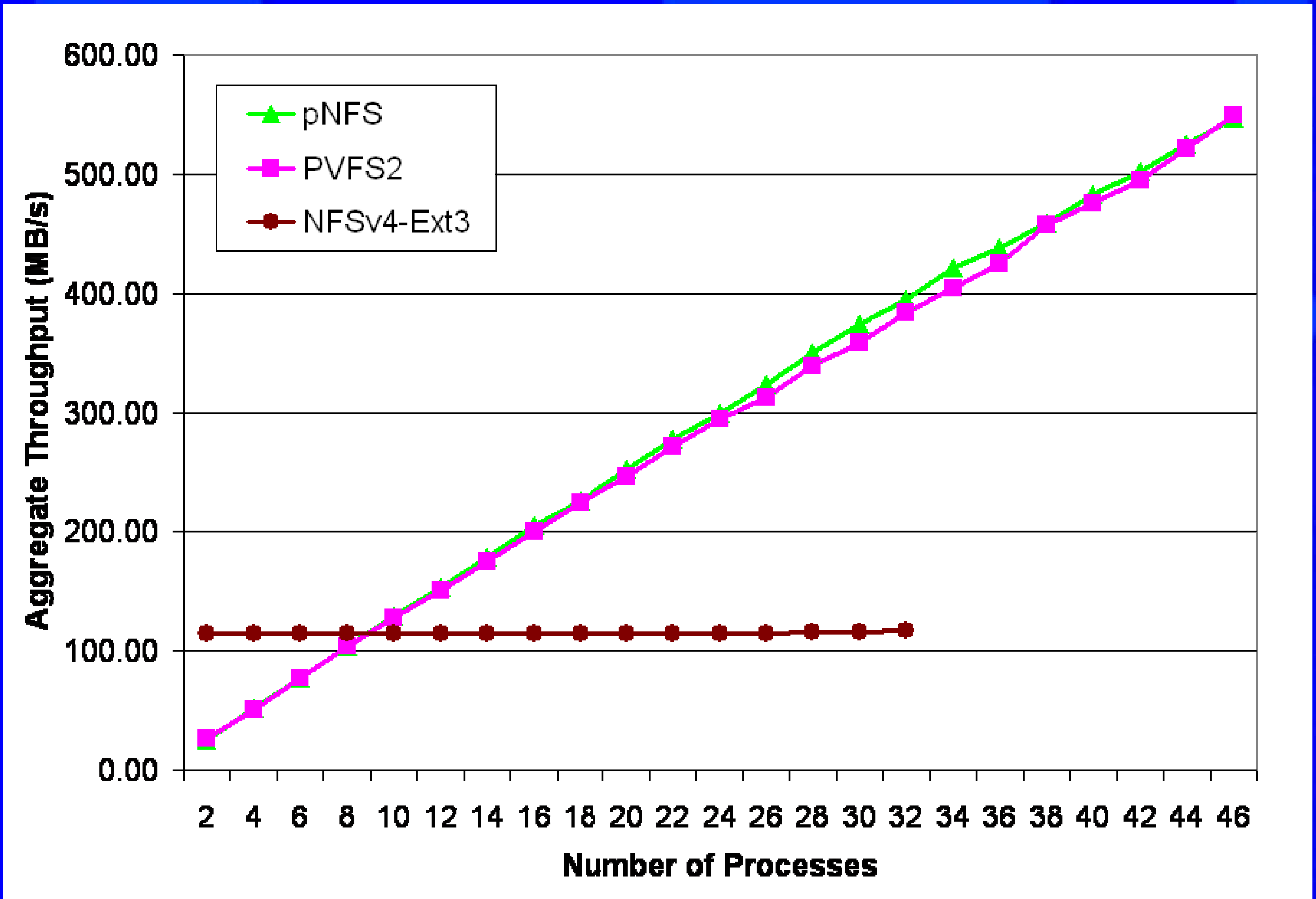
pNFS/PVFS2 Prototype Architecture



Write Experiments - Separate Files



Read Experiment - Separate Files



pNFS Prototype Extensions

- Optional client data cache
- Client writeback cache
- Client read and write request gathering

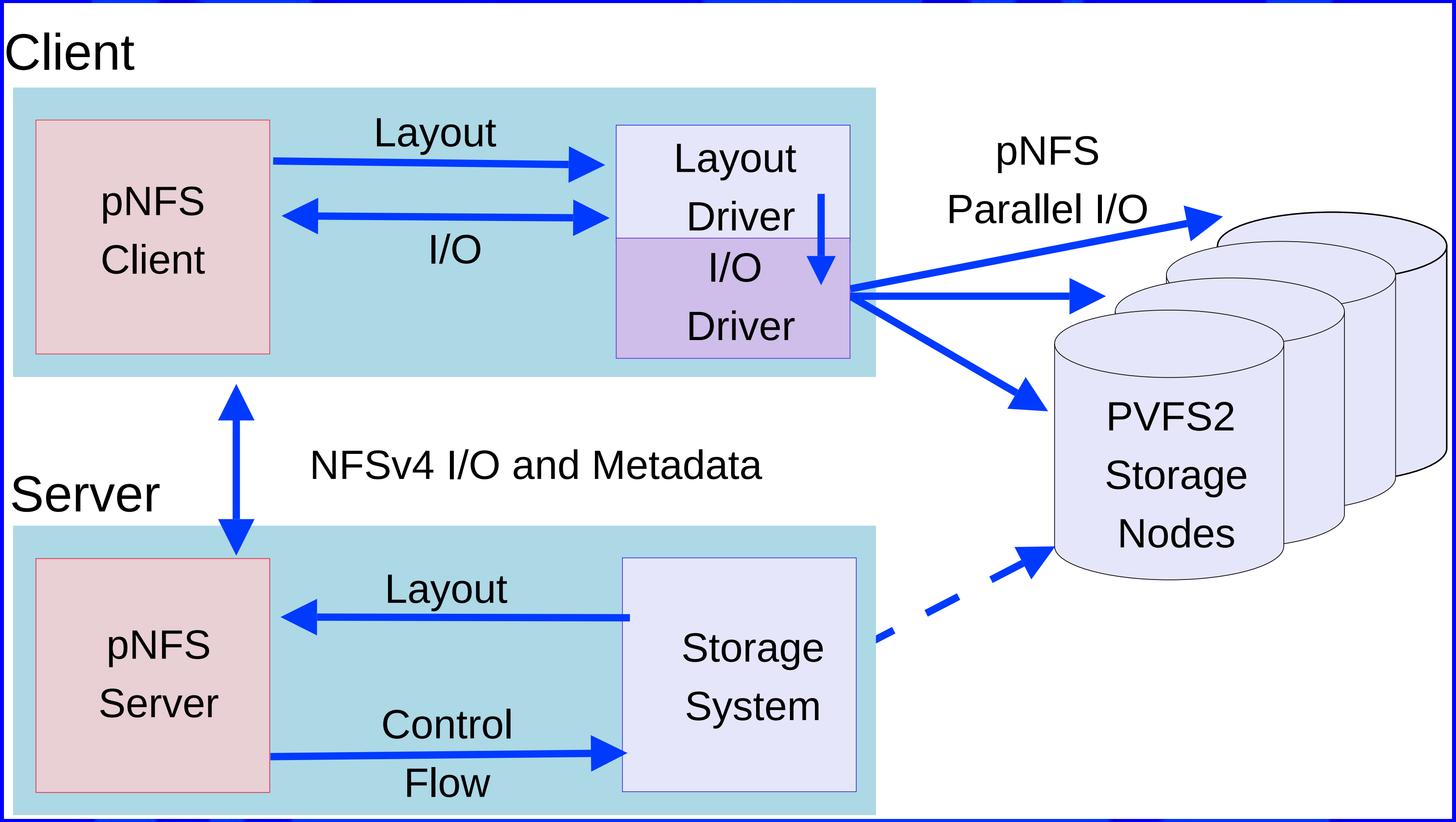
More Information

- pNFS/PVFS2 Paper and Prototype –
  www.citi.umich.edu/projects/asci
- pNFS –
  www.pdl.cmu.edu/pNFS
- NFSv4 –
  www.nfsv4.org

NFSv4 Protocol

- Fully integrated security with RPCSEC_GSS
- Integrate Mount and Lock protocols
- File delegations
  - Client assumes ownership of file
- OPEN and CLOSE commands
  - Server state
- Enhanced cross-platform interoperability
- Framework for file system migration and replication
- Request bundling
- NFSv4.1
  - Sessions and RDMA
  - Directory delegations
  - Secinfo

pNFS Architecture



PVFS2 and BlueGene

- PVFS2 features:
  - Largely in userspace
  - Stateless clients; very tolerant of client failures
    - No locks, leases, caches, etc to revalidate/reacquire
  - Good support for heterogeneous clients/servers
- BlueGene features:
  - Thousands of lightweight clients
    - Reboot after every job
  - 3 architectures: storage (x86), login (ppc64), support (ppc32) nodes
- Turns out PVFS2 fits pretty well
  - Honestly, not a surprise – but we're still happy

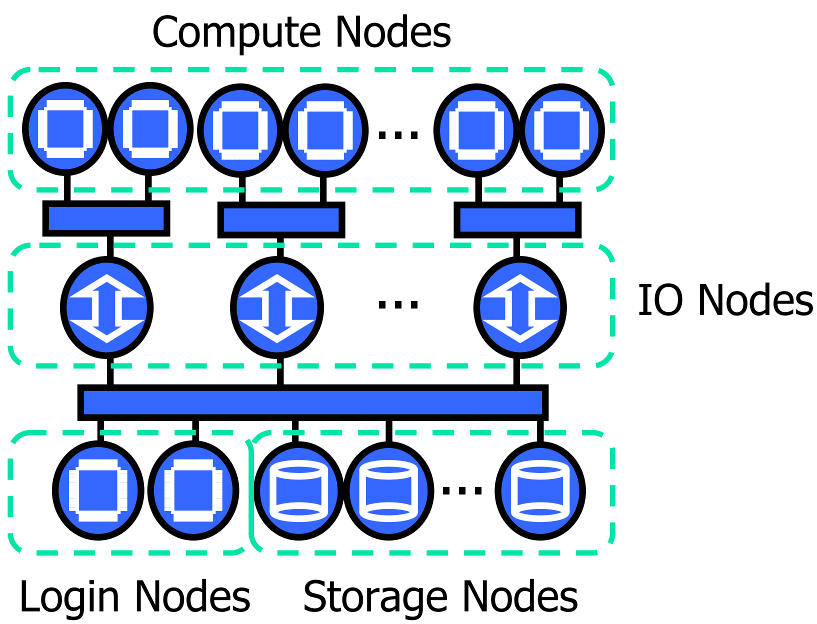
PVFS2 and ZeptoOS

- ZeptoOS: researching operating systems for petascale architectures
- ZeptoOS group at ANL:
  - Pete Beckman
  - Kamil Iskra
  - Kazutomo Yoshii
  - Susan Coghlan

  http://www.mcs.anl.gov/zeptoos/
- Guess which file system they use?
- For BGL, ZeptoOS provides custom ramdisk, nicely integrated PVFS2 support

View of I/O on BG/L

- Storage nodes
  - Local access to disks
  - GigE connections to login and IO nodes
- Login nodes
  - Interactive machines
  - <u>Place where data staging will occur</u>
- Support nodes (IO node)
  - Aggregators for compute node I/O
    - 1:8 to 1:64 ratio of IO nodes to compute nodes
  - Tree connection to compute nodes
- Compute nodes
  - <u>Source/sink of runtime I/O</u>



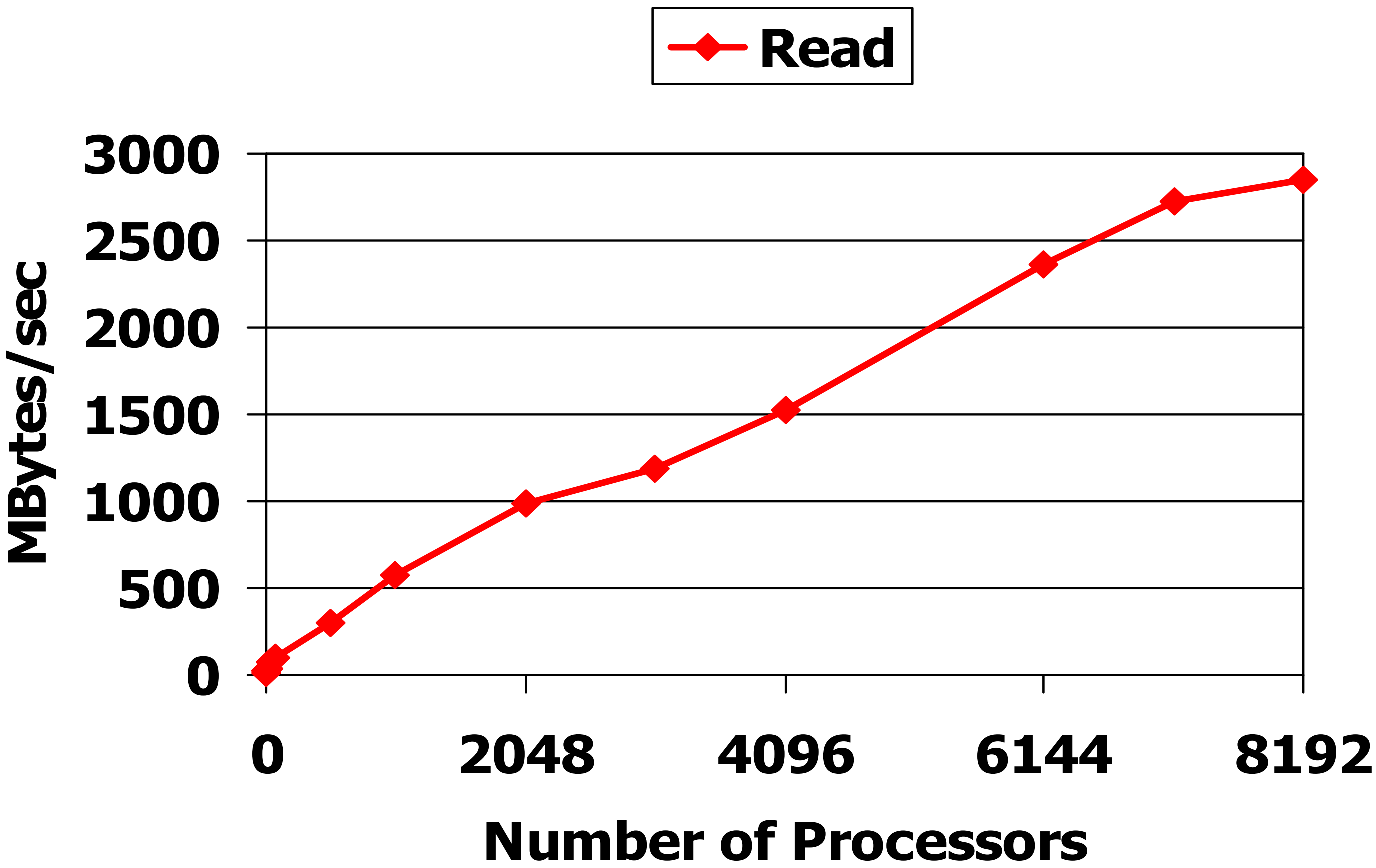
Porting PVFS2

- Rough Timeline:
  - Feb 7: figure out how to use BGL (compiling, running simple MPI jobs)
  - Feb 15: PVFS2 running on servers, login nodes
  - Feb 18: ANL gets source to support node kernel
  - Feb 21: Collect first full-rack scalability numbers
- From dead stop to running I/O benchmarks in 2 man-weeks

BGL and PVFS2 performance at large scale

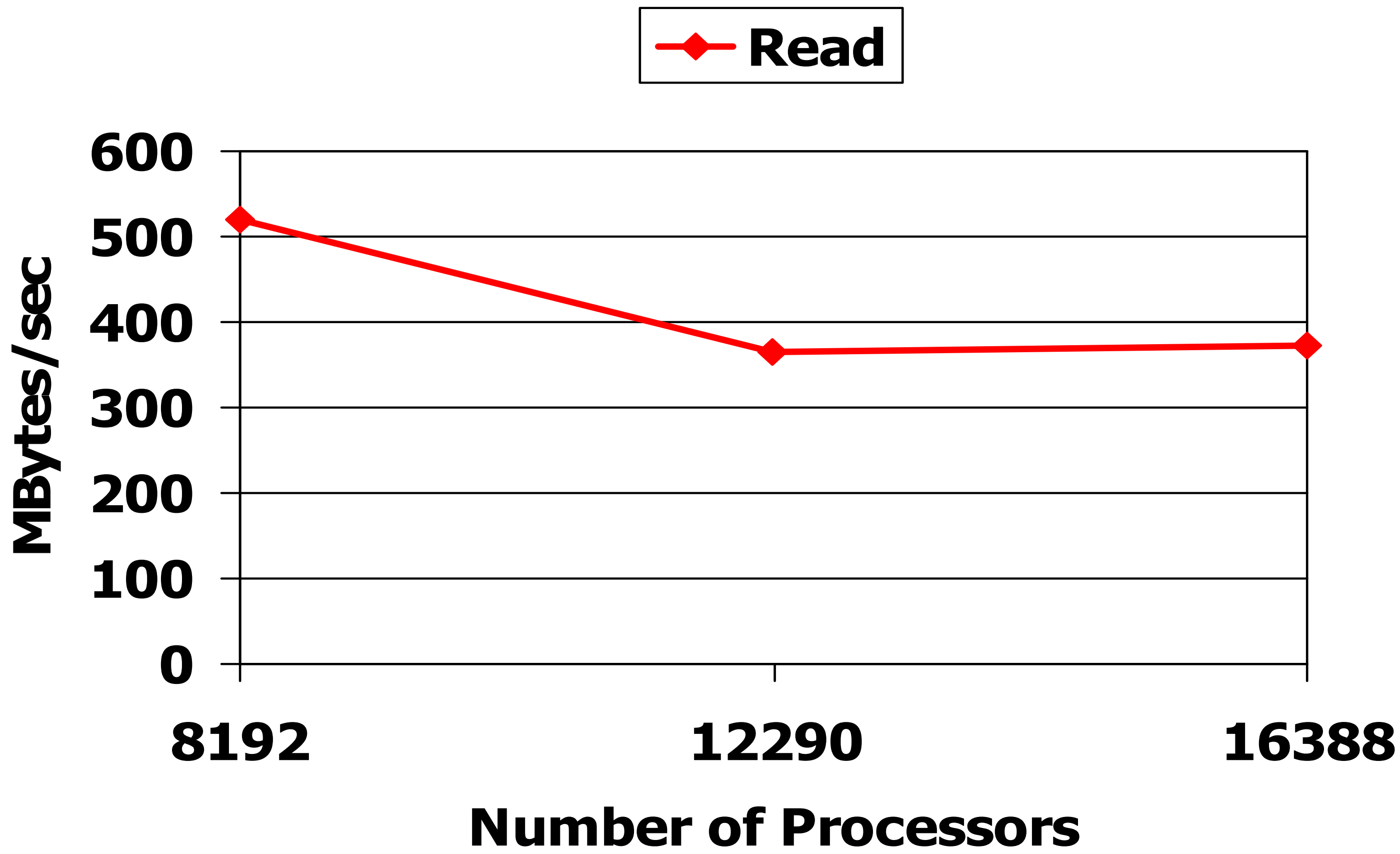
- BGW:
  - large BlueGene installation at IBM Watson
  - 20480 compute nodes
  - 33 storage nodes (PVFS2 servers)
  - 320 support (IO) nodes: 1 per 64 compute
- Recently held “BGW day”
  - Give app and library developers several hours on 16k nodes.
  - IBM allowed us to install PVFS2
    - Yeah, surprised us too. Thanks, IBM!

BGW read perf, up to 8K nodes



- 33 PVFS2 servers: 86 MB/sec per PVFS2 server
- 64 compute nodes per IO node: 22 MB/sec per support node

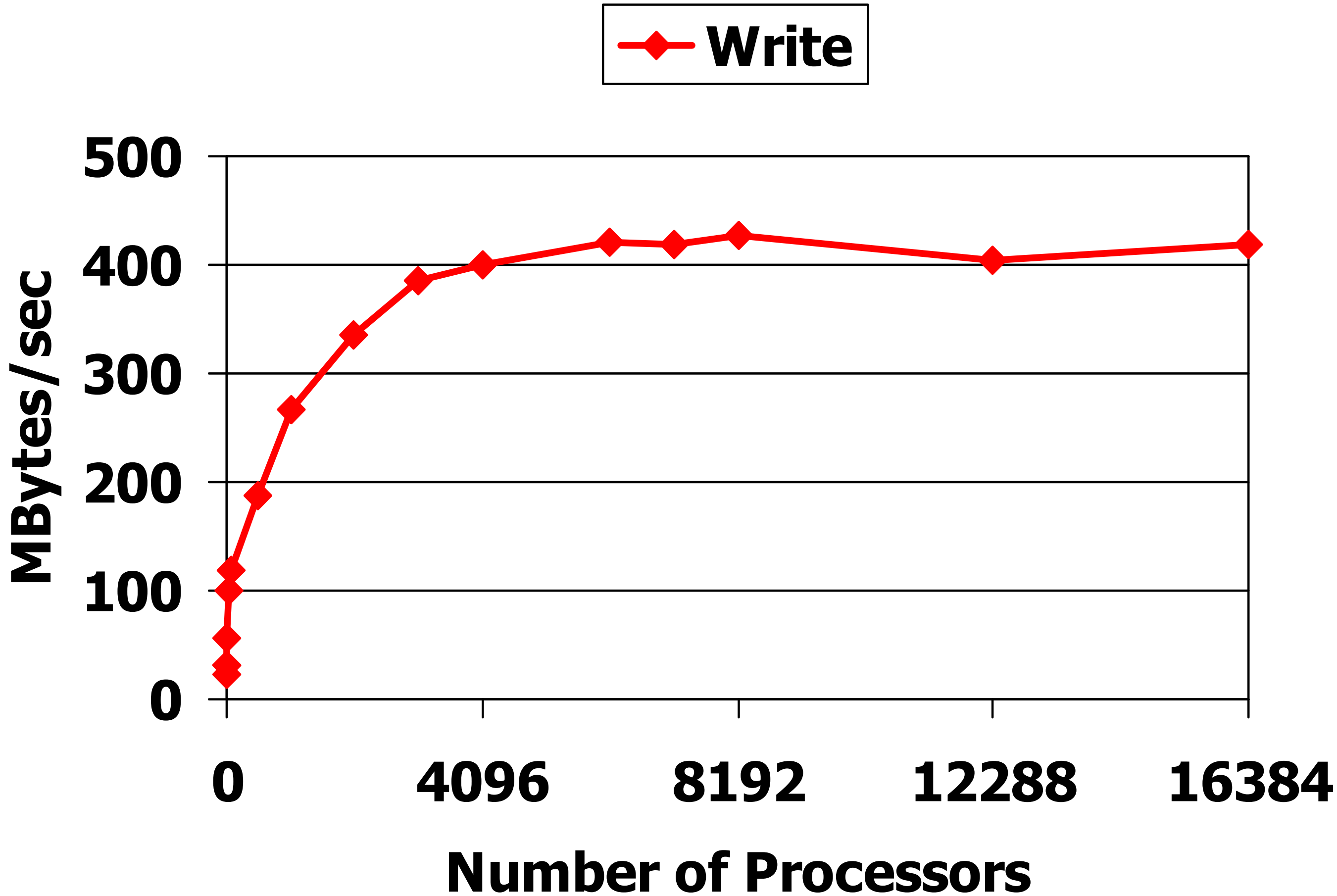
BGW read performance (2 of 2)



- different hw config than prior slide
- Bad performance probably due to Ethernet

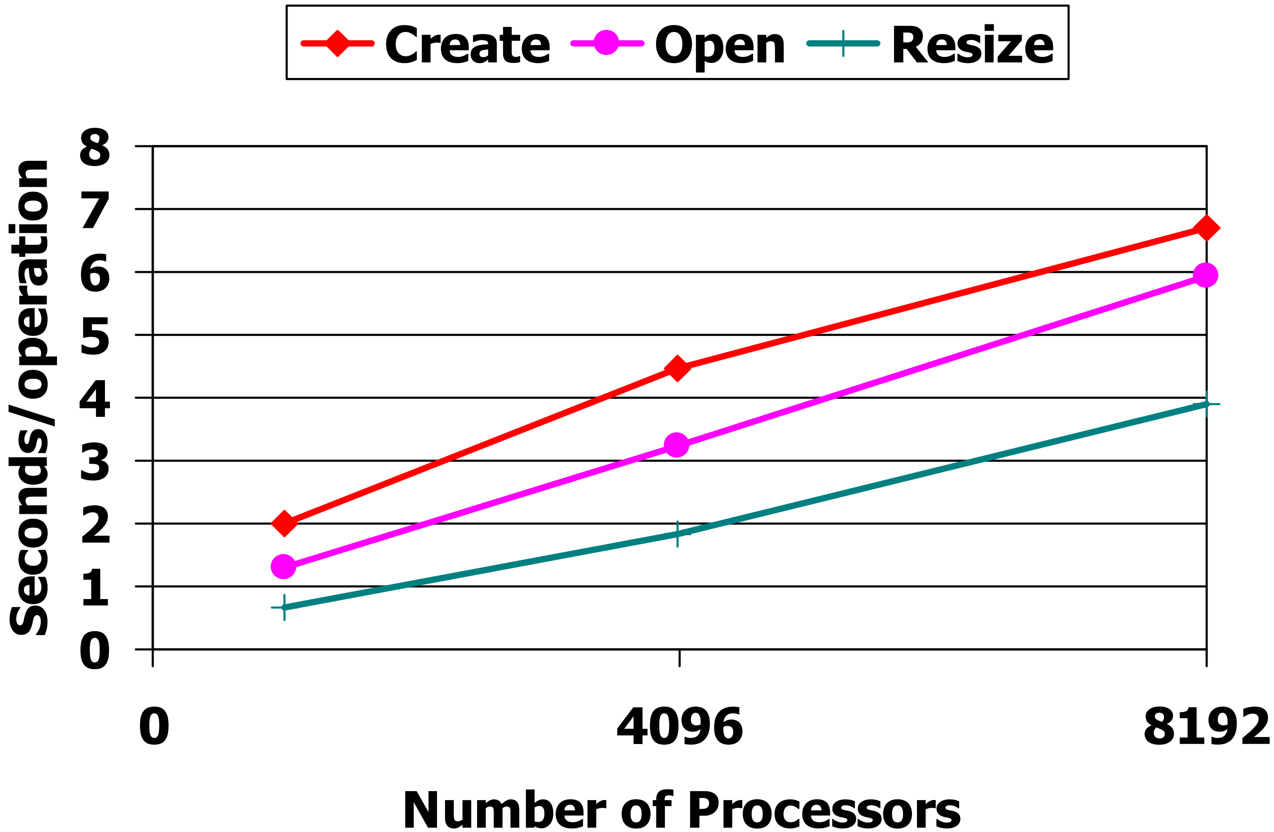
http://www.pvfs.org/pvfs2

BGW write performance



- Sync after write: ciod serialization
- ciod effects much larger than job placement

BGW MPI-IO metadata



- Have to use UFS MPI-IO implementation
- Already looking at ways to improve BGL infrastructure. Compare this next year!

Everything Else

- Lots of additional work in progress:
  - Small I/O optimizations (Argonne)
    - Think eager mode for I/O
  - Extented attributes (Argonne)
    - Researching new ways to direct file system behavior
  - Metadata tuning (Heidelberg)
    - Experimenting with new trove implementations
  - Data mirroring (Clemson)
    - Won't be anything left for users to ask of us!

Additional information

- PVFS2 web site: http://www.pvfs.org/pvfs2
  - Documentation, mailing list archives, and downloads
- PVFS2 mailing lists (see web site)
  - Separate users and developers lists
  - Please use these for general questions and discussion!
- Internet Relay Chat (IRC)
  - Server irc.freenode.net, channel #pvfs2
  - Talk directly with developers
- Email
  - Rob Ross <rross@mcs.anl.gov>
  - Walt Ligon <walt@clemson.edu>
  - Pete Wyckoff <pw@osc.edu>
  - Rob Latham <robl@mcs.anl.gov>
  - Sam Lang <slang@mcs.anl.gov>

Thanks for coming!

Any Questions?

http://www.pvfs.org/pvfs2